**LeetCode-ByteDance**

**3. Longest Substring Without Repeating Characters**

用哈希表记录每个字符出现的位置，用双指针指向不重复串的开始和结束，每次移动结束的指针，如果遇到相同的字符，则后移开始的指针

**46. Permutations**

DFS，递归实现，为了加速，每次选出来的元素放在最后

（最快的方法应该是数组直接定长，然后传参决定每次位置，比较复杂，暂时没有做）

**47. Permutations II**

DFS，递归实现，由于有重复，用dict记录每个数字出现的次数，每次拿出一个数，把对应的次数减一之后递归即可

**105. Construct Binary Tree from Preorder and Inorder Traversal**

前序遍历的第一个节点是根节点，利用这个可以把中序遍历分为两段，第一段是左子树，中间是根节点，右边是右子树。然后递归调用即可

**112. Path Sum**

直接dfs，递归实现

**141. Linked List Cycle**

快慢指针，一个一步，一个两步，相遇就是有环

**146. LRU Cache**

用hashmap加双向链表

**155. Min Stack**

用栈将当前数值和最小值一起存即可

**179. Largest Number**

定制排序，按照第一位的大小排，如果相同看第二次，依次，但是难实现

改为直接将两个数相加，然后交换位置相加，比较相加后的结果，与原来的要求相同

Python代码的简洁写法

<https://leetcode.com/problems/largest-number/solution/>